REPORT NO. 38

CONTRACT N-123(61756)19425A/PMR

REARRANGEMENT OF SUBPROGRAMS ON TAPE TO REDUCE SELECTION TIME

by Werner E. Knolle, Ph.D.

LAND-AIR, INC.
PARADYN DIVISION
POINT MUGU, CALIFORNIA
A SUBSIDIARY OF
DYNALECTRON CORPORATION

TEST DATA DIVISION
Range Operations Department
Pacific Missile Range

23 October 1962

This report was prepared under contract number N-123(61756)19425A/PMR in support of the Test Data Division Computing service. The program described herein, enables the realization of optimum machine time through program arrangement on the library tape.

Copies are available to interested personnel, on request, from the Test Data Division, Range Operations Department, Code 3280, Pacific Missile Range.

Approved: [illegible]
R. M. CONWAY, Manager
Paradyn Division
Land-Air, Inc.

Approved: Walter L. Milne
WALTER L. MILNE
Contract Tech Coordinator
Test Data Division

Approved: Charles J. Thorne
CHARLES J. THORNE
Head, Test Data Division
Range Operations Department

ABSTRACT

A number of subprograms are read from a magnetic tape into the storage of an electronic computer during the execution of a computing job. The order in which these subprograms are arranged on the tape is being studied. A number of jobs are executed during a certain period of time using different groups of subprograms from the tape. It is our aim to re-arrange the subprograms on the tape in order to reduce the time it takes to read the subprograms into storage.

FOREWORD

An expression for the total time of selection of the subprograms is determined. It is not the aim of this paper to find the ideal distribution in order to minimize the selection time. Instead a method of trial and error is used to improve the distribution sufficiently for practical purposes. Exchanges between two subprograms at a time are made. If an exchange does not reduce the time of selection a re-exchange is made. A program in FORTRAN for the IBM 7090 is supplied to complete the task. A practical example is given.

TABLE OF CONTENTS

I SELECTION OF SUBPROGRAMS

A subprogram on magnetic tape consists of a number of records with record gaps between them. Each record contains a certain number of "words". A word has a fixed length on the tape. Hence, the length of a subprogram may be measured in terms of the number of words it contains within each record, allowing for record gaps. Let us suppose we have n subprograms $s_1, s_2, \ldots, s_n$ of lengths $t_1, t_2, \ldots, t_n$ words on the tape. The "head", which reads the words from the tape, is located at the beginning of the tape, and this is also the beginning of s_1. To select the subprogram s_m $(m > 1)$ the head moves over $(t_1 + t_2 + \ldots, + t_{m-1})$ words. After the first subprogram s_m has been read into storage, the head is located at the beginning of s_{m+1}. To select the next subprogram s_r, the head moves over

a) $(t_{m+1} + t_{m+2} + \ldots, + t_{r-1})$ words if $r > m + 1$

b) $(t_m + t_{m-1} + \ldots, + t_r)$ words if $r < m$.

It takes no time to select s_{m+1} $(r = m + 1)$. If the same subprogram is used again $(r = m)$ we can ignore the corresponding selection time, since it cannot be reduced by changing the distribution. In this fashion, the head moves forwards and backwards over the tape to select all the subprograms for a job.

We assume that the head moves forwards and backwards over the tape with the same constant speed. Hence, the total selection time may be measured in terms of the number of words passed by the head without reading.

TOTAL TIME OF SELECTION

There are n subprograms $s_i (i = 1,\ldots,n)$ on the tape of lengths $t_i (i = 1,\ldots,n)$. During a certain period of time, a number of jobs have been executed and the numbers $f_{ik} (i,k = 1,2,\ldots,n)$ and $f_i (i = 1,2,\ldots,n)$ have been recorded. f_i is the number of times the subprogram s_i was selected first during the execution of a job. f_{ik} is the number of times the selection of s_i was followed by the selection of s_k. To find the total time of selection, T, we form the following expressions:

$$f_i (t_1 + t_2 + \ldots, + t_{i-1}),\ i > 1$$

$$f_{ik} (t_{i+1} + t_{i+2} + \ldots, + t_{k-2} + t_{k-1}),\ k > i + 1$$

$$f_{ki} (t_i + t_{i+1} + \ldots, + t_{k-1} + t_k),\ k > i$$

and find the sum of all possible terms:

$$T = \sum_{i=2}^{n} f_i \sum_{m=1}^{i-1} t_m + \sum_{i=1}^{n-2} \sum_{k=i+2}^{n} f_{ik} \sum_{m=i+1}^{k-1} t_m + \sum_{k=2}^{n} \sum_{i=1}^{k-1} f_{ki} \sum_{m=i}^{k} t_m .$$

We change this expression for faster computation. The third term may be changed to

$$\sum_{k=2}^{n} f_{k1} \sum_{m=1}^{k} t_m + \sum_{i=2}^{n-1} f_{n_i} \sum_{m=i}^{n} t_m + \sum_{k=3}^{n-1} \sum_{i=2}^{k-1} f_{ki} \sum_{m=i}^{k} t_m .$$

The second term may be changed to

$$\sum_{i=1}^{n-2} f_{i,i+2} \cdot t_{i+1} + \sum_{i=1}^{n-3} \sum_{k=i+3}^{n} f_{ik} \sum_{m=i+1}^{k-1} t_m .$$

The first term may be written as

$$f_2 \cdot t_1 + \sum_{k=2}^{n-1} f_{k+1} \sum_{m=1}^{k} t_m .$$

Hence,

$$T \doteq f_2 \cdot t_1 + \sum_{k=2}^{n-1} (f_{k1} + f_{k+1}) \sum_{m=1}^{k} t_m + f_{n1} \sum_{m=1}^{n} t_m$$

$$+ \sum_{i=2}^{n-1} f_{ni} \sum_{m=i}^{n} t_m + \sum_{i=2}^{n-1} f_{i-1,i+1} \cdot t_i$$

$$+ \sum_{k=3}^{n-1} \sum_{i=3}^{k} f_{k,i-1} \sum_{m=i-1}^{k} t_m + \sum_{i=3}^{n-1} \sum_{k=i}^{n-1} f_{i-2,k+1} \sum_{m=i-1}^{k} t_m .$$

The last two terms add up to

$$\sum_{i=3}^{n-1} \sum_{k=i}^{n-1} (f_{k,i-1} + f_{i-2,k+1}) \sum_{m=i-1}^{k} t_m .$$

This form of T has been used in the subroutine FINDT to compute T.

III REDUCTION OF THE TIME OF SELECTION

To reduce the time of selection T we change the order in which the subprograms are on the tape. We compute at first T_0 for the original distribution. Then we exchange two subprograms s_i and s_k and compute T_1. If $T_1 \geq T_0$, we exchange s_i and s_k again. Otherwise we make the next exchange. We let $i = 1,2,\ldots,n-1$ and $k = i + 1,\ i + 2,\ldots,n$ until a permanent change has been made. Then we start again with $i = 1$ and $k = 2$. This method will lead to a distribution which cannot be improved by the exchange of any two subprograms. For $n > 50$ it may take hours of computer time before the final distribution has been found, because the changes become more and more insignificant and harder to find as time goes on. It is advisable to run the program for 10 minutes at a time and break it off as soon as the improvement becomes insignificant. In an example with $n = 65$ the time T was reduced by 26% during the first 10 minutes of computer time, by 7% during the next 30 minutes run, by 4% during the next 30 minutes run and by 1% during the next 20 minutes run. In an example with $n = 71$ the time T was reduced by 5.5% in 7 minutes and by 1.6% in another 30 minutes. In this case the program worked on an improved distribution of subprograms.

To enable a restart of the program without a change of the input data f_{ik}, f_i, and t_i, the program transforms the matrices f_{ik}, f_i, and t_i to any given order of distribution $s_a, s_b, s_c, \ldots, s_x, s_y, s_z$. We simply type the integers a, b, c,...,x, y, z on certain data cards, and the program establishes this order and continues making improvements.

IV A PROGRAM IN FORTRAN

A program in FORTRAN for the IBM 7090 is provided. The arrays F(N,N), T(N) and S(N) contain the matrices f_{ij}, t_i and f_i, where N = n. The array MU(N) contains the integers a, b, c,...,x, y, z mentioned in 3. The array NU(N) is initially made to agree with MU(N) and then subjected to changes, indicating the current order of distribution,

$S_{NU(1)}, S_{NU(2)}, \ldots, S_{NU(N)}$.

The total time of selection is denoted by TT, the change in time due to an exchange of subprograms by DT. The following subroutines are used;

READ: This subroutine reads from the data cards the arrays T,F,MU and S. F must be typed columnwise on the cards, e.q. $f_{11}, f_{21}, \ldots, f_{n1}, f_{12}, f_{22}, \ldots f_{n2}, \ldots$, where $f_{11} = f_{22} = \ldots, f_{nn} = 0$. MU must contain the numbers 1, 2, 3,..., N at the start and later the improved order of distribution.

EXCH: This subroutine exchanges the subprograms S_{L1} and S_{L2}. It makes the necessary changes in the matrices F,T,NU and S.

HEAD: This subroutine prints the total time TT and the order of distribution NU as a heading for each page.

ORDER: This subroutine prints the total time TT and the order of distribution NU whenever 10 changes have been made.

INIT: This subroutine transforms to the order of distribution NU = MU.

FINDT: This subroutine evaluates TT.

PLAN: This subroutine exchanges two subprograms at a time and prints for each permanent change the numbers L1, L2 and DT, indicating that the L1th and L2nd subprograms have been exchanged and that TT has been increased by DT.

The main program calls INIT,PLAN,ORDER and EXIT.
N is taken to be 71 in the following listing of the program.

ORDER

```
      MAIN PROGRAM
    1 CALL INIT
    2 CALL PLAN
    3 CALL ORDER
    4 CALL EXIT
      END(1,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
      SUBROUTINE INIT
      DIMENSION F(71,71),T(71),NU(71),MU(71),S(71),LU(71)
      COMMON F,T,TT,DT,NU,LN,N,L1,L2,MU,S
    1 N=71
    2 CALL READ
   17 DO 19 I=1,N
   18 NU(I)=I
   19 LU(I)=I
   20 N1=N-1
   21 DO 27 I=1,N1
   22 L1=I
   23 L2=MU(I)
   24 L2=LU(L2)
   25 IF(L2-L1)24,27,26
   26 CALL EXCH
   27 LU(I)=L2
   28 CALL FINDT
   31 CALL HEAD
      RETURN
      END(1,0,0,0,0,0,1,0,0,0,0,0,0,0,0)
      SUBROUTINE EXCH
      DIMENSION F(71,71),T(71),NU(71),MU(71),S(71)
      COMMON F,T,TT,DT,NU,LN,N,L1,L2,MU,S
    5 DO6 I=1,N
      A=F(L1,I)
      F(L1,I)=F(L2,I)
    6 F(L2,I)=A
    7 DO10 I=1,N
    8 A=F(I,L1)
    9 F(I,L1)=F(I,L2)
   10 F(I,L2)=A
   11 A=T(L1)
   12 T(L1)=T(L2)
   13 T(L2)=A
   14 A=NU(L1)
   15 NU(L1)=NU(L2)
   16 NU(L2)=A
   17 A=S(L1)
   18 S(L1)=S(L2)
   19 S(L2)=A
      RETURN
      END(1,0,0,0,0,0,1,0,0,0,0,0,0,0,0)
      SUBROUTINE READ
      DIMENSION F(71,71),T(71),NU(71),MU(71),S(71)
      COMMON F,T,TT,DT,NU,LN,N,L1,L2,MU,S
    1 READ INPUT TAPE 5,2,T
    2 FORMAT(16F5.0)
    3 READ INPUT TAPE 5,4,F
    4 FORMAT(24F3.0/24F3.0/23F3.0)
    5 READ INPUT TAPE 5,6,MU
    6 FORMAT(40I2)
    7 READ INPUT TAPE 5,4,S
      RETURN
      END(1,0,0,0,0,0,1,0,0,0,0,0,0,0,0)
      SUBROUTINE FINDT
      DIMENSION F(71,71),T(71),NU(71),MU(71),S(71)
      COMMON F,T, TT,DT,NU,LN,N,L1,L2,MU,S
    1 N1=N-1
    2 TT=0.0
    3 A=T(1)
      TT=TT+A*S(2)
    4 DO 6 I=2,N1
    5 A=A+T(I)
    6 TT=TT+A*(F(I,1)+S(I+1))
      A=A+T(N)
      TT=TT+A*F(N,1)
    7 DO 9 I=2,N1
    8 A=A-T(I-1)
    9 TT=TT+A*F(N,I)
   10 DO 14 I=3,N1
   11 A=T(I-1)
   12 DO 14 J=1,N1
   13 A=A+T(J)
   14 TT=TT+A*(F(I-2,J+1)+F(J,I-1))
   15 DO 16 I=2,N1
   16 TT=TT+T(I)*F(I-1,I+1)
   17 RETURN
      END(1,0,0,0,0,0,1,0,0,0,0,0,0,0,0)
      SUBROUTINE HEAD
      DIMENSION F(71,71),T(71),NU(71),MU(71),S(71)
      COMMON F,T,TT,DT,NU,LN,N,L1,L2,MU,S
    1 WRITE OUTPUT TAPE 6,2,TT,NU
    2 FORMAT(1H1,9X,5HTIME=,E16.8,12HTHE ORDER IS/(1X,I2,39I3))
    3 LN=0
    4 RETURN
      END(1,0,0,0,0,0,1,0,0,0,0,0,0,0,0)
      SUBROUTINE PLAN
      DIMENSION F(71,71),T(71),NU(71),MU(71),S(71)
      COMMON F,T,TT,DT,NU,LN,N,L1,L2,MU,S
      A=TT
    1 N2=N-1
    2 DO 16 I=1,N2
      L1=I
    3 L3=L1+1
    4 DO 16 J=L3,N
      L2=J
    5 CALL EXCH
      CALL FINDT
      DT=TT-A
    6 IF(DT)7,16,16
    7 WRITE OUTPUT TAPE 6,9,L1,L2,DT
    9 FORMAT(10X,10HCHANGE L1=,I4,8H AND L2=,I4,4H DT=,E16.8)
      A=TT
   10 LN=LN+1
   11 IF(50-LN)12,12,13
   12 CALL HEAD
   13 IF(XMODF(LN,10))2,14,2
   14 CALL ORDER
   15 GO TO 2
   16 CALL EXCH
   17 RETURN
      END(1,0,0,0,0,0,1,0,0,0,0,0,0,0,0)
      SUBROUTINE ORDER
      DIMENSION F(71,71),T(71),NU(71),MU(71),S(71)
      COMMON F,T,TT,DT,NU,LN,N,L1,L2,MU,S
    1 WRITE OUTPUT TAPE 6,2,TT,NU
    2 FORMAT(8X,13HNEW ORDER.TT=,E16.8/(1X,I2,39I3))
      LN=LN+1
      RETURN
      END(1,0,0,0,0,0,1,0,0,0,0,0,0,0,0)
```